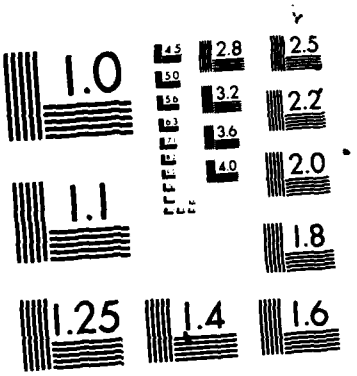
AD-A172 776 AN INTERATIVE METHOD FOR COMPUTING BIVARIATE C SUB 1 1/1
PIECEWISE CUBIC POLY (U) WISCONSIN UNIV-MADISON
MATHEMATICS RESEARCH CENTER T A GRANDINE JUN 86
UNCLASSIFIED MRC-TSR-2937 DAAG29-80-C-0041 F/G 12/1 NL



MRC Technical Summary Report #2937

AN ITERATIVE METHOD FOR COMPUTING BIVARIATE C^1 PIECEWISE CUBIC POLYNOMIAL INTERPOLANTS

Thomas A. Grandine

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

June 1986

(Received June 3, 1986)

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

National Science Foundation
Washington, DC 20550

UNIVERSITY OF WISCONSIN — MADISON
MATHEMATICS RESEARCH CENTER

AN ITERATIVE METHOD FOR COMPUTING BIVARIATE C^1 PIECEWISE CUBIC POLYNOMIAL INTERPOLANTS

Thomas A. Grandine[1,2]

Technical Summary Report #2937

June 1986

ABSTRACT

This paper describes a successive overrelaxation (SOR) method for computing a class of bivariate C^1 piecewise cubic polynomial interpolants. Given a collection of points in R^2 together with a triangulation of those points, the schemes described require only the values of the function to be interpolated at the given points. The result is a C^1 interpolant which is a cubic polynomial over each of the triangles in the triangulation. Numerical results are presented for a typical scheme in this class.

AMS(MOS) Subject Classifications: 41A15, 41A63, 65D05, 65F10

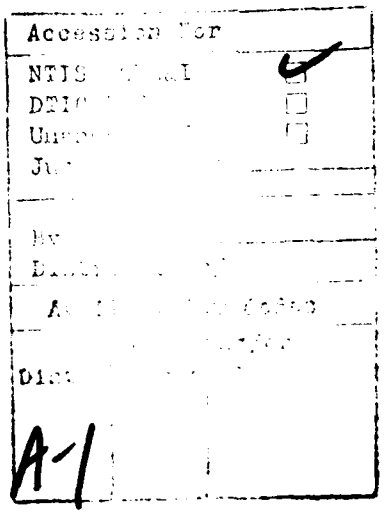
Key Words: Iterative method, SOR, piecewise polynomial. B-form of a polynomial

Work Unit Number 3 - Numerical Analysis and Scientific Computing

[1]Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.
[2]This material is based upon work supported by the National Science Foundation under Grant No. DMS-8210950, Mod. 4.

SIGNIFICANCE AND EXPLANATION

This paper describes a class of bivariate C^1 piecewise cubic polynomial interpolation schemes for scattered data. Computation of this interpolant requires the solution of a linear system of equations which is extremely sparse. Since the triangulation of scattered data may allow for any number of triangles to meet at a vertex, the matrix corresponding to this linear system need not have a nice structure. This means that the ordering of the vertices of the triangulation may be of critical importance whenever a direct method of solving the linear system is to be employed. In this paper, an iterative method of solving this system is proposed in order to eliminate this difficulty. Numerical evidence suggests that this iterative scheme is an extremely efficient and effective way of solving the linear system.



The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

AN ITERATIVE METHOD FOR COMPUTING BIVARIATE C^1 PIECEWISE CUBIC POLYNOMIAL INTERPOLANTS

Thomas A. Grandine[1,2]

Interpolation of bivariate scattered data by smooth piecewise polynomials is a problem which has received a great deal of attention in recent years ([A84-1], [A84-2], [A84-3], [BF81], [BL84], [F80], [F83], [L77], [L84], [Z70], and elsewhere). The case of C^1 cubics (piecewise cubic polynomials with one continuous derivative) has received a good deal of this attention. In this paper, a method of cheaply constructing such interpolants is given. The idea is to take any old piecewise cubic interpolant (which need not be C^1) and compute a perturbation of it which satisfies the desired smoothness conditions.

The method outlined in this paper is based on the B-form representation of bivariate polynomials, as given in [B86] and [F80]. Readers not familiar with the B-form should read [B86] for a detailed description of it in its most general form. In any case, the features of the bivariate B-form required to establish the schemes in this paper will be reviewed. See [B86] for proofs of these facts.

Consider the points u, v, $w \in \mathbf{R}^2$. An arbitrary point $x \in \mathbf{R}^2$ can be expressed as an affine combination of u, v, and w, i.e. $x = \alpha_u(x)u + \alpha_v(x)v + \alpha_w(x)w$, where $\alpha_u(x) + \alpha_v(x) + \alpha_w(x) = 1$. The numbers $\alpha_u(x)$, $\alpha_v(x)$, and $\alpha_w(x)$ are called the **barycentric coordinates** of x with respect to u, v, and w, and they are, in fact, affine functions of their

[1]Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.
[2]This material is based upon work supported by the National Science Foundation under Grant No. DMS-8210950, Mod. 4.

argument. Denote by α the vector of barycentric coordinates, and consider the multi-index $i \in \mathbf{Z}_+^3$ with $i_u + i_v + i_w = n$. It is clear that

$$\alpha^i := \frac{\alpha_u^{i_u} \alpha_v^{i_v} \alpha_w^{i_w}}{i_u! i_v! i_w!} \tag{1}$$

is a bivariate polynomial of degree at most n (since each component of α is affine). The $\binom{n+2}{n}$ valid choices for i give rise to the same number of linearly independent bivariate polynomials of degree $\leq n$. Since the dimension of the space of bivariate polynomials of degree $\leq n$ is also $\binom{n+2}{n}$, the functions α^i must form a basis for that space. Thus, any polynomial p of degree $\leq n$ can be written in the form

$$p = \sum_i c_i \alpha^i,$$

for certain coefficients c_i. This is the **B-form** of a bivariate polynomial.

This form is so useful because it has a nice geometric interpretation. Consider the triangle given as the convex hull of u, v, and w. Then all the valid choices of i can be viewed as occupying locations on a certain mesh over the triangle. This mesh is the one generated by considering all of the points in the triangle with barycentric coordinates i/n, for each of the valid choices of i. For the case $n = 3$, the locations are given in Figure 1. For example, the point corresponding to u itself has $i_u = 3$, $i_v = i_w = 0$. The point nearest u along the edge from u to v has $i_u = 2$, $i_v = 1$, and $i_w = 0$. The point in the center has $i_u = i_v = i_w = 1$.

This form has the nice property that the values of the coefficients say something about the local behavior of p. For example, if $i_u = n$, with $i_v = i_w = 0$, then $c_i = p(u)$, i.e. the coefficient associated the each of the vertices is just the value of the polynomial at

that vertex. Moreover, if the restriction of p to the line determined by one of the edges is considered, then that univariate polynomial is uniquely determined by the coefficients associated with that edge. These properties make it possible to discuss representations of piecewise polynomial functions using a generalization of the B-form, namely the **B-net**.

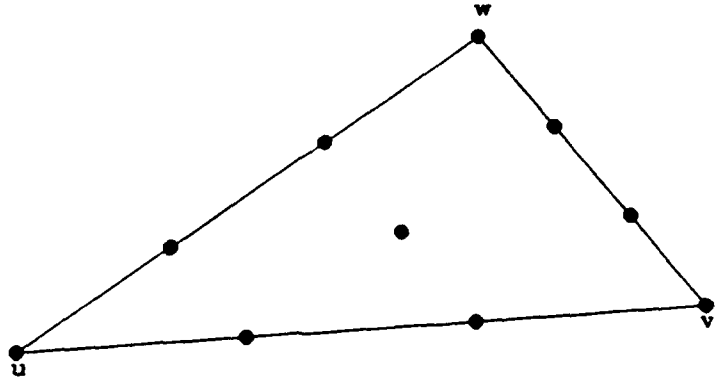


Figure 1

Consider more generally 4 points in $\mathbf{R}^2$, say u, v, w, and z, whose convex hull is a quadrilateral. Suppose that this quadrilateral is divided into two triangles, one of which is the convex hull of u, v, and w, and the other is the convex hull of v, w, and z. Let the quadrilateral be the domain for some piecewise polynomial function of degree $\leq n$ with two pieces, each of which has as its domain one of the triangles. It is clear that each polynomial may be described in B-form on each triangle. If, as is generally the case, the pp function is continuous, then the univariate polynomials which are the restrictions of each polynomial to the common edge, in this case given by v and w, must be the same. Since these polynomials are determined by the B-form coefficients along those edges, and

the polynomials must be the same, the coefficients must also be the same. The B-form representation of continuous pp functions is called the B-net. The B-net for the above example, with $n = 3$, is given in Figure 2.

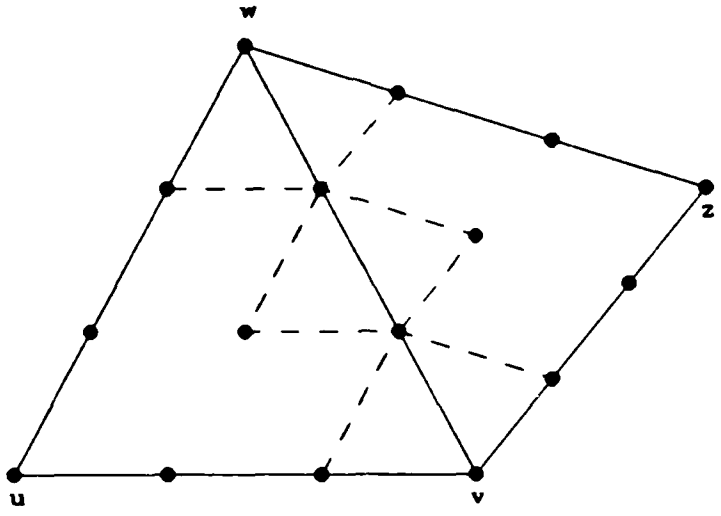


Figure 2

In this paper, the goal is to construct C^1 interpolants with cubic pp functions. Thus, simple continuity of pp functions is not enough. Fortunately, the conditions for C^1 smoothness in terms of the B-net are both simple and elegant. In Figure 2 there are three quadrilaterals marked with a dashed line, each of which is similar to the domain. For the pp function given to be C^1, it is necessary and sufficient that for each small quadrilateral, the four B-net coefficients associated with its vertices satisfy a certain linear relationship: If the four coefficients are viewed as function values at the points u, v, w, and z, related by the similarity of the small quadrilaterals to the big one, then the graph of the four points

must be planar. In general, n of these conditions will be required for pp functions of degree $\leq n$.

To turn now to the main topic of the paper, consider any collection of points in $\mathbf{R}^2$ at which the values of some function which is to be interpolated are given. Assume that a triangulation of these points is also given. If not, one can always be computed by the method outlined in [GS78]. The so-called Delauney triangulation produced by this method is viewed by many as being a good one for the purpose of interpolation, and it has been shown ([Si78]) to be the same triangulation as that produced by a very different method in [L72]. In any case, if there are d points in the interior of this triangulation and e on the boundary, then there will be $2d + e - 2$ triangles in the triangulation [BL84]. From this fact, it follows that there are $3d + 2e - 3$ edges in the triangulation, of which $3d + e - 3$ are interior edges.

Given this collection of points, consider piecewise cubic polynomial functions defined over the triangulation, i.e. functions which are locally cubic polynomials in each triangle. The B-net for these functions consists of $9d + 6e - 8$ coefficients. However, not all of these coefficients are variables in the situation considered here. Only those functions which actually interpolate need to be considered. This uniquely determines the coefficients located at the vertices, leaving only $8d + 5e - 8$ variables. Making the interpolant $\mathbf{C}^1$ amounts to imposing three linear conditions at each interior edge, for a total of $9d + 3e - 9$ equations. Thus, computing a bivariate $\mathbf{C}^1$ cubic pp interpolant for scattered data boils down to solving the linear system

$$Ax = b \tag{2}$$

for some $9d+3e-9$ by $8d+5e-8$ matrix A and an $8d+5e-8$ vector b, both of which depend upon the location of the vertices and the structure of the triangulation. The vector b also depends upon the function values which are to be interpolated. The smoothness conditions are homogeneous except when one of the four coefficients is known. This happens exactly when one of the four points is a vertex of the triangulation.

A great deal is known about A and b. It is known, for example, that A is extremely sparse, having either three or four non-zero entries in each row and no more than three non-zero entries in each column. Exactly one third of the entries in b are zero, and these correspond to the rows in A which have exactly four non-zero entries. It is also known ([MS77], [S79], and [S84]) that the row rank of A is no larger than $7d + e - 9$, which guarantees that the linear system (2) is underdetermined. It also means that (2) might not be solvable. As of this writing, the solvability of (2) remains a conjecture. No one has been able to prove the existence of a solution or to produce a set of points (and a triangulation) for which there is none. For the remainder of this paper, the existence of a solution to the linear system (2) will be assumed.

In order to construct $\mathbf{C}^1$ cubic pp interpolants to the data, some way of eliminating the extra $d+4e+1$ degrees of freedom which arise because of the rank deficiency of A is needed. One way of doing this is to begin by ignoring the linear system (2) and to construct any old interpolant. For example, the piecewise linear interpolant may be considered. It is very easy to construct, since the value of each of the cubic B-net coefficients is simply the value of the piecewise linear interpolant at that point ([B86] and [F80]). In any case, suppose that some interpolant is given, and let $\hat{x}$ be the vector of B-net coefficients corresponding to it. It is clear that, in general, $A\hat{x} \neq b$, since A and b were never considered in forming

$\hat{x}$. However, this does provide a way of eliminating the extra degrees of freedom inherent in (2). A C^1 piecewise cubic interpolant to the given data can be computed by solving

$$\begin{aligned} &\min \|e\|_2 \\ &\text{subject to } Ae = r, \end{aligned} \tag{3}$$

where $r := b - A\hat{x}$, and A and b are as before. If e solves (3), then $A(\hat{x} + e) = b$, so solving (3) amounts to computing a perturbation of the original interpolant in order to make it C^1.

The problem two may be solved by many different techniques. One way, for example, would be to compute the singular value decomposition of A. While this method has many highly desirable numerical properties, it is inappropriate here because it fails to take advantage of the sparsity of the matrix A. The same is true of Lemke's method and other pivotal methods, which, even if they make use of sparse matrix techniques, will still depend in some critical way on the ordering of the rows and columns of A. This in turn will depend upon how the vertices of the triangulation (and the triangles themselves) happen to be ordered. In this paper, this difficult question will be overlooked by using an iterative method, namely successive overrelaxation (SOR), to solve the problem.

The space in which e lies can be decomposed into two orthogonal subspaces, namely $\ker(A)$, the nullspace of A, and $\mathrm{ran}(A^T)$, the range of A^T. Any solution to $Ae = r$ has a unique decomposition $e = e_k + e_r$, where $e_k \in \ker(A)$ and $e_r \in \mathrm{ran}(A^T)$. In order to obtain a least norm solution to $Ae = r$, i.e., to have e solve (3), it is necessary and sufficient to have $e_k = 0$. Thus, if $e := A^T y$, then solutions to

$$AA^T y = r \tag{4}$$

will also provide solutions to (3).

Let L be the matrix the same size as AA^T whose entries are the same as AA^T below the main diagonal and zero everywhere else. More straightforwardly, L is the strict lower triangular part of AA^T. Let D be the diagonal part of AA^T, so that $L + D + L^T = AA^T$. It is clear that D has strictly positive entries on its main diagonal, since each of these entries is just the inner product of a row of A with itself. Then the iterative scheme

$$y^{n+1} = y^n + \omega D^{-1}(r - Ly^{n+1} - Dy^n - L^T y^n) \tag{5}$$

converges for any choice of $0 < \omega < 2$. Here y^n is the vector which is the n-th iterate of the scheme, and the convergence is to any solution of the system (4). Since A is rank deficient, it is clear that AA^T, although symmetric and positive semi-definite, may nevertheless be singular. The convergence is a consequence of the following theorem, due to Keller:

Theorem 1 [K65]: Let S be a symmetric matrix of order m and let N be a non-singular matrix of order m for which

$$P := N + N^T - S$$

is positive definite. Then the iterative scheme

$$Nx^{n+1} = (N - S)x^n + b \tag{6}$$

converges if and only if S is positive semi-definite and $b \in \mathrm{ran}(S)$. The convergence is to a solution of $Sx = b$.

The iterative scheme (5) is of the form (6) if N is chosen to be $L + \frac{1}{\omega}D$. For any finite choice of ω, this is non-singular. Then

$$\begin{aligned} P &= N + N^T - AA^T \\ &= (L + \frac{1}{\omega}D) + (L + \frac{1}{\omega}D)^T - (L + D + L^T) \\ &= \frac{2-\omega}{\omega}D, \end{aligned}$$

which is positive definite if $0 < \omega < 2$. Hence, SOR converges to some solution of the linear system (4).

As it turns out, it doesn't matter which solution to (4) is obtained. Any solution y has two orthogonal components, $y_k \in \ker(AA^T)$ and $y_r \in \mathrm{ran}((AA^T)^T) = \mathrm{ran}(AA^T)$, so that $y = y_k + y_r$. However, no matter what y_k is, $A^T y_k = 0$, since $A^T y_k \in \mathrm{ran}(A^T)$ and $A^T y_k \in \ker(A)$. Thus, $e = A^T y$ depends only on y_r, and not on y_k. This uniquely solves the system (3).

The only remaining step in the interpolation process is to take e and add it to $\widehat{x}$. This will provide B-net coefficients which satisfy the conditions for $\mathbf{C}^1$ smoothness, interpolate the given data, and are, in the 2-norm sense, as close as possible to those in the original interpolant. Since the polynomials (1) are a well-conditioned basis for the space of pp functions considered [H82], it is reasonable to expect that the resulting interpolant will also, in some sense, be as close as possible to the original interpolant. This means that if the original interpolant was constructed to satisfy certain shape requirements, it is at least conceivable that the resulting $\mathbf{C}^1$ interpolant may satisfy those same requirements.

Before proceeding to the numerical experiments, there are a few features of this scheme that are worth noting. The first is that the matrix AA^T is extremely sparse. Since each row of A has at most four non-zero elements, each row (and column) of AA^T has at most

eight non-zero elements in addition to the entry on the main diagonal. Because it is also symmetric only half ot it needs to be stored in memory in any computer implementation of the scheme.

It is also worth noting that the assumed solvability of the linear system (2) is an essential ingredient in Theorem 1. Without this, the iterative scheme cannot converge. Since this is the only condition in the theorem which is not ironclad, the appearance of a problem for which the method fails to converge provides a candidate for a counter-example to the conjecture. In all experiments performed so far, non-convergence has never been observed.

The numerical experiments which have been performed have all used a local implementation of the scheme (for piecewise linear initial-guess interpolants) on a VAX-11/780 running VMS. This implementation is in the form of a FORTRAN subroutine which takes as input an array of points in $\mathbf{R}^2$ and an array of corresponding function values. It begins by calling a Delauney triangulation routine to determine a suitable triangulation of the region of interpolation. Next, it determines the piecewise linear interpolant for the given data, i.e. the vector $\hat{x}$. After forming the matrix A and the vector b, the matrix AA^T and the vector $r = b - A\hat{x}$ are computed. SOR is then used to solve the system $AA^T y = r$ to a tolerance specified by the user with the parameter ω also specified by the user. Finally, the vector $\hat{x} + A^T y$ is computed, and its components are put into an array which contains the B-net for the interpolant.

One of the most important features for any interpolant to have is that it be **local**, which means that a change in one of the function values will only change the resulting interpolant near the location of the function value. This property has been investigated

for a number of different data sets by choosing all of the function values to be zero except at one point where it is chosen to be one. The graphs of the resulting interpolants are all very similar to the one shown in Figure 3. It is clear from this that the interpolant is not local, but might be essentially local, meaning that the effects of a change in a function value decay with distance.

If, in fact, the interpolant is essentially local, the error in the interpolant should be at least as good as $O(\rho^2)$, where ρ is the maximum **diameter** of any of the triangles in the triangulation. Here, diameter means the length of the longest segment which is contained within a triangle. The error estimate should be this good because the interpolation scheme used **reproduces** linear polynomials. In other words, if the given data describes a linear function, then the interpolant will be precisely that function.

For the bivariate function $f(x_1, x_2) = x_1^2 + x_2^2 - 2x_1x_2 + x_1 + 2x_2 + 3$ defined over the square $[0,1]^2$, interpolants on successively finer triangulations have been computed. In each case, the interpolation points are located at the vertices of an $m \times m$ square mesh placed over $[0,1]^2$. The following table summarizes the results. The column labelled k is the observed rate of convergence between each entry and the subsequent one. Here k is the exponent if the error is assumed to behave like $O(\rho^k)$. The errors listed are the maximum absolute differences between the function and its interpolant which are encountered.

m	Error	k
3	0.14779282	1.939
4	0.06732988	1.989
5	0.03799295	1.994

6	0.02434874	1.994
7	0.01692724	1.998
8	0.01243925	1.998
9	0.00952625	2.015
10	0.00751376	2.135
12	0.00489593	1.831
14	0.00360560	1.993
16	0.00271082	

The observed rate of convergence is very nearly the $O(\rho^2)$ which had been hypothesized. It would have been shocking, in fact, to have obtained anything better, since the function is quadratic and is clearly not being reproduced by the scheme!

Also deserving of mention is the fact that the $m = 16$ case, which is interpolation of 256 function values with 450 (perhaps) different bivariate cubic polynomials, required less than 321.94 seconds to compute. Given that this involved the triangulation of the 256 points (a needless waste, since the mesh is regular anyway), the construction and solution of a 1935×1935 linear system of equations, and the evaluation of the interpolant on a 51×51 grid (2601 interpolant evaluations), this seems quite reasonable. Certainly much larger problems are possible even on the VAX.

Of course, the method works well independent of the regularity of the triangulation. Consider the function $g(x_1, x_2) = (x_1 - x_1^2)(x_2 - x_2^2)e^{3x_1^2 - 7x_2^2}$. A contour map of this function over the square $[0,1]^2$ is given in Figure 4. For this function, its interpolant has been computed over the triangulation given in Figure 5. This triangulation is the Delauney

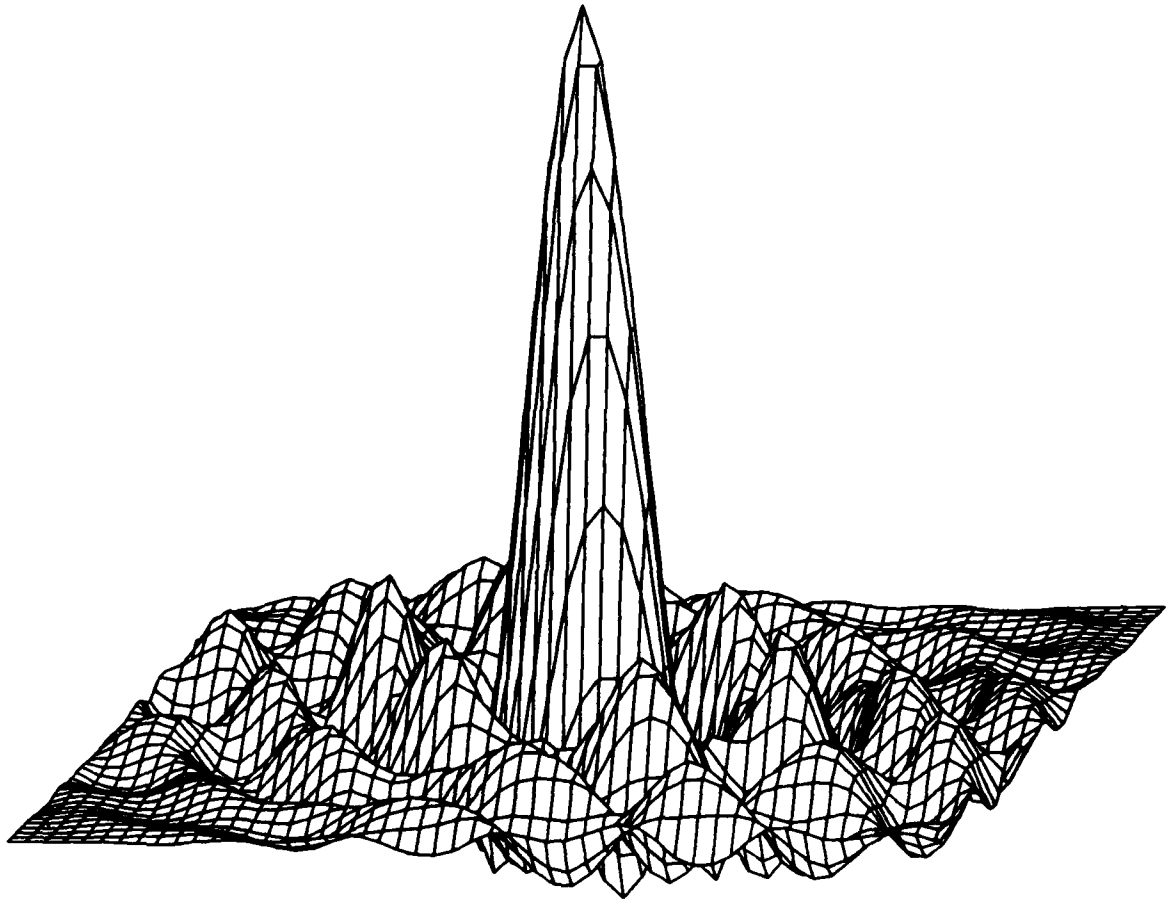


Figure 3

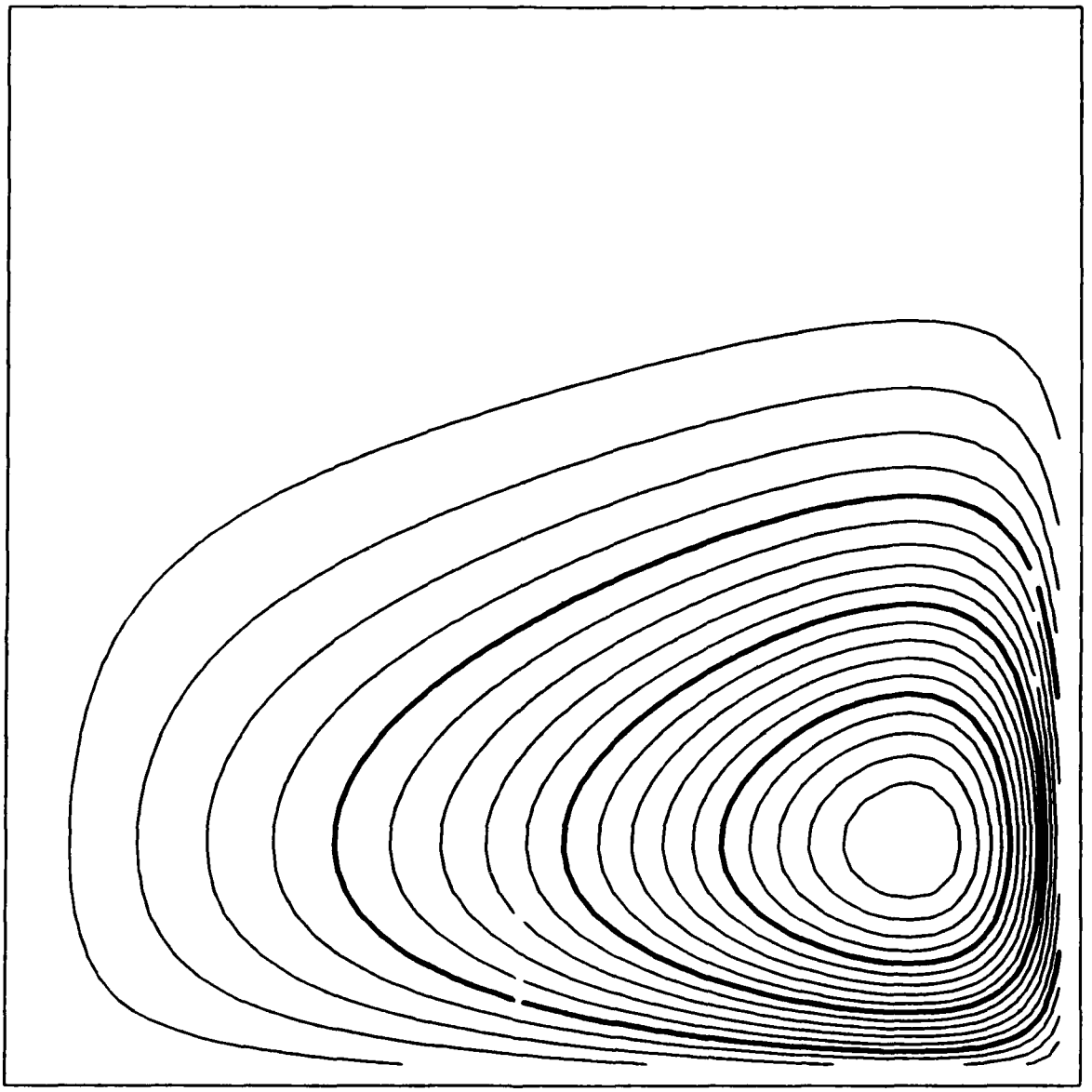


Figure 4

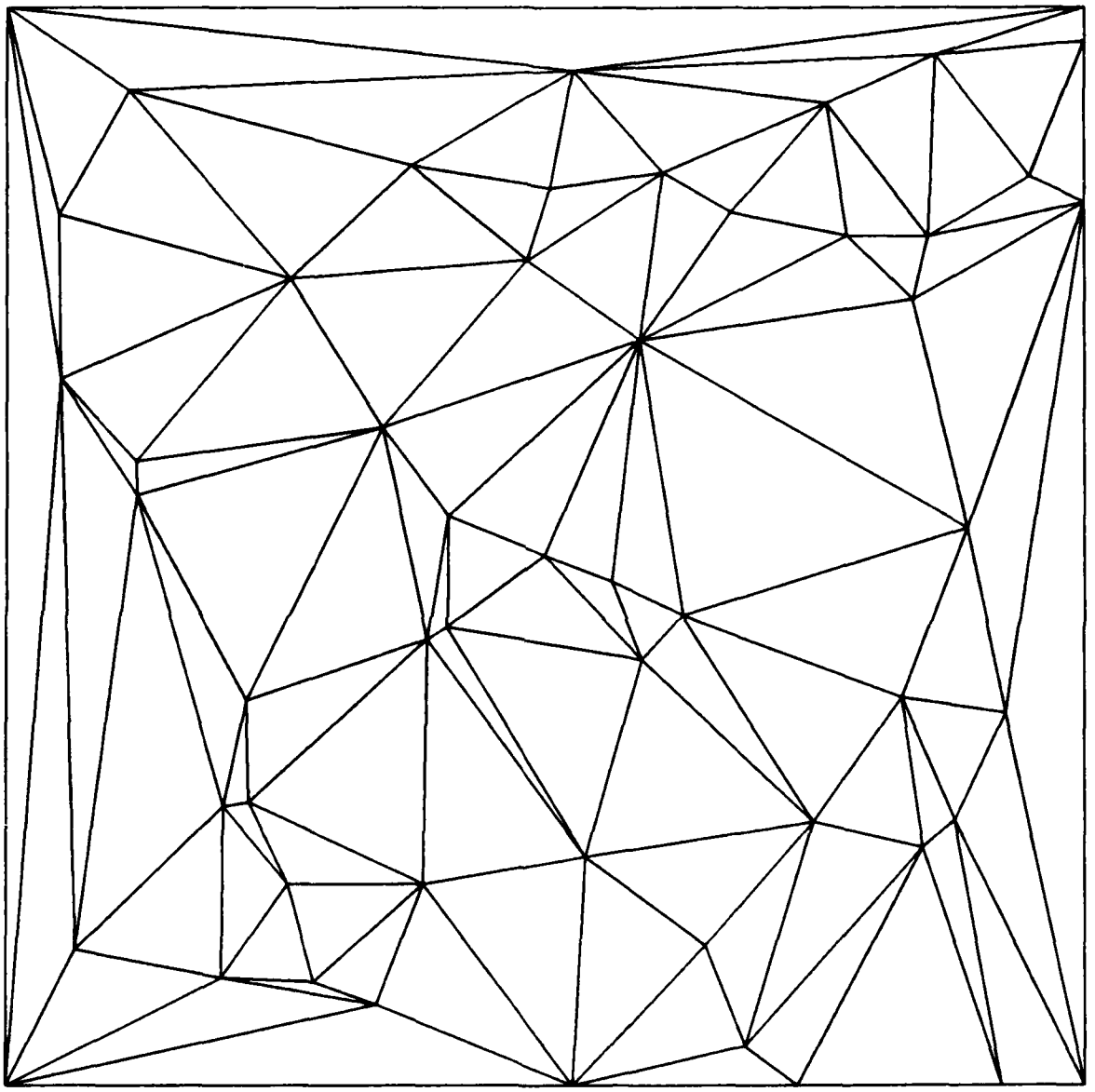


Figure 5

triangulation for a collection of 54 random points. (Actually, only 50 of them are random. The four vertices of the unit square were thrown in for luck.) A contour map of the interpolant is given in Figure 6, and a surface plot of the error function is given in Figure 7. The maximum difference between the function and its interpolant is 0.021338049.

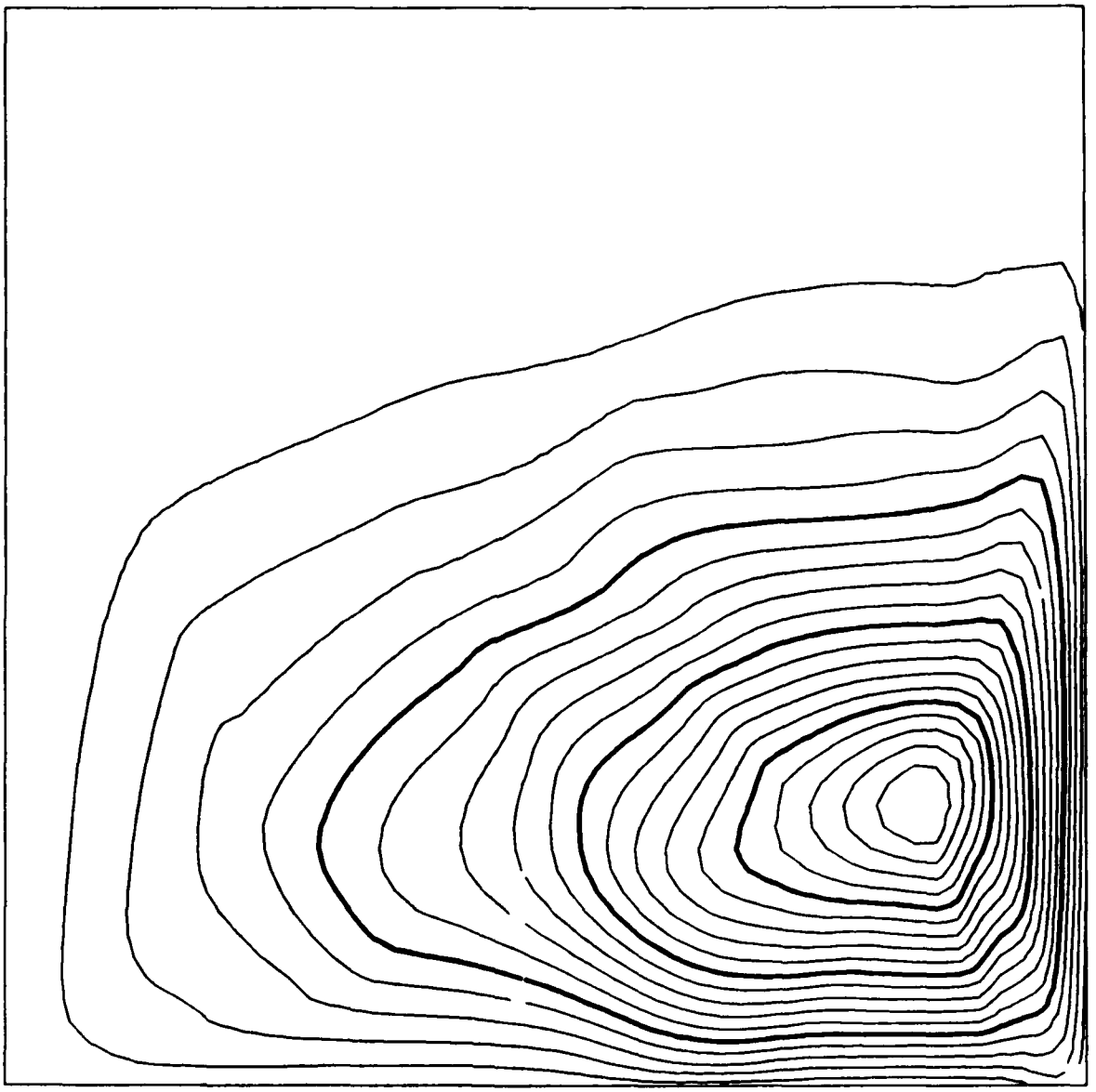


Figure 6

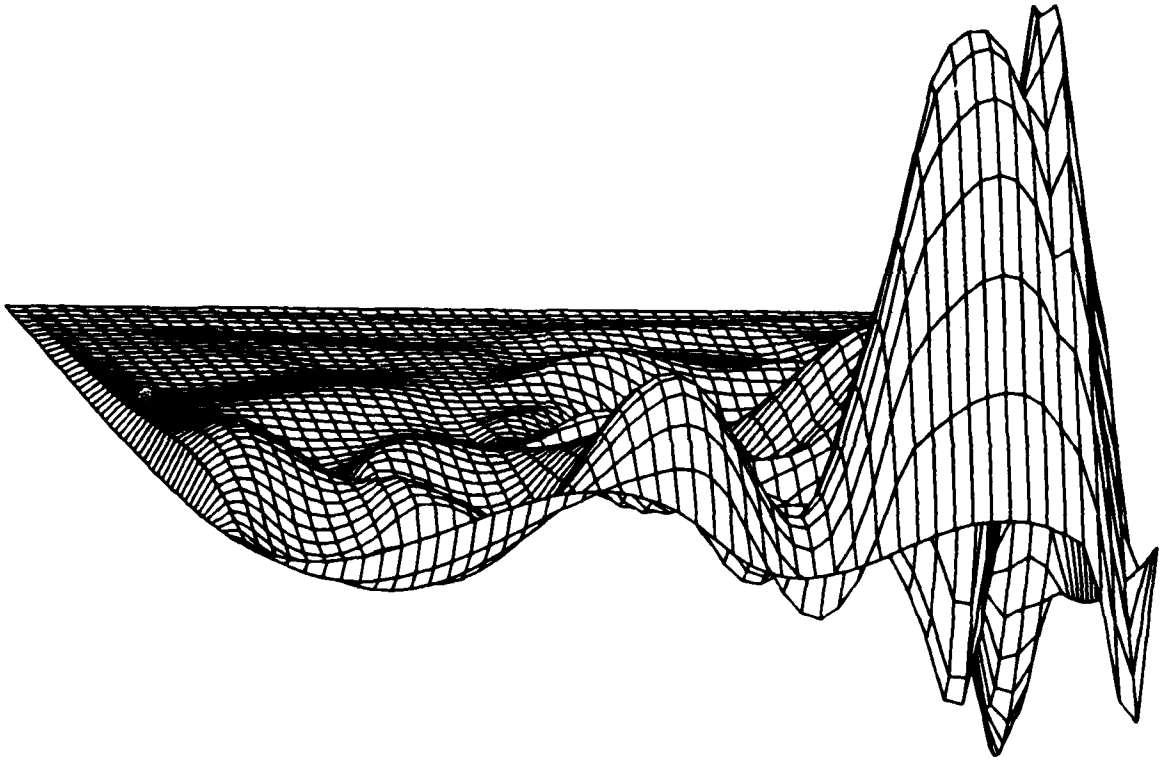


Figure 7

References

[A84-1] Alfeld, Peter. "A Discrete C^1 Interpolant for Tetrahedral Data," *Rocky Mountain Journal of Mathematics* **14**, pp. 5–16.

[A84-2] Alfeld, Peter. "A Bivariate C^2 Clough–Tocher Scheme," *Computer Aided Geometric Design* **1**, pp. 257–267.

[A84-3] Alfeld, Peter. "A Trivariate Clough–Tocher Scheme for Tetrahedral Data," *Computer Aided Geometric Design* **1**, pp. 169–181.

[BF81] Barnhill, R. E. and Farin, G. "C^1 Quintic Interpolation over Triangles: Two Explicit Representations," *Int. J. for Num. Meth. in Eng.* **17**, pp. 1763–1778.

[BL84] Barnhill, R. E. and Little, F. F. "Three- and Four-Dimensional Surfaces," *Rocky Mountain Journal of Mathematics* **14**, pp. 77–102.

[B86] de Boor, Carl. "B-Form Basics," in preparation, 1986.

[F80] Farin, Gerald. "Bezier Polynomials over Triangles and the Construction of Piecewise C^r Polynomials," TR/91, Department of Mathematics, Brunel University, Uxbridge, Middlesex, UK, 1980.

[F83] Farin, Gerald. "Smooth Interpolation to Scattered 3D Data," in *Surfaces in Computer Aided Geometric Design*, R. E. Barnhill and W. Böhm, eds., North Holland, 1983.

[GS78] Green, P. J., and Sibson, R. "Computing Dirichlet Tesselations in the Plane," *The Computer Journal* **21**, pp. 168–173.

[H82] Höllig, Klaus. "Multivariate Splines," *SIAM Journal of Numerical Analysis* **19**, pp. 1013–1031.

[K65] Keller, H. B. "On the Solution of Singular and Semidefinite Linear Systems by Iteration," *J. SIAM Numer. Anal. Ser. B* **2**, pp. 281–290.

[L72] Lawson, C. L. "Generation of a Triangular Grid with Application to Contour Plotting," California Institute of Technology Jet Propulsion Laboratory, Technical Memorandum 299, 1972.

[L77] Lawson, C. L. "Software for C^1 Surface Interpolation," in *Mathematical Software II*, J. R. Rice, ed., Academic Press: New York, pp. 161–194.

[L84] Lawson, C. L. "C^1 Surface Interpolations for Scattered Data on a Sphere," *Rocky Mountain Journal of Mathematics* **14**.

[MS77] Morgan, J. and Scott, R. "The Dimension of Piecewise Polynomials," manuscript, 1977.

[S79] Schumaker, L. L. "On the Dimension of Spaces of Piecewise Polynomials in Two Variables," in *Multivariate Approximation Theory*, W. Schempp and K. Zeller, eds., Basel: Birkhäuser, pp. 396–412.

[S84] Schumaker, L. L. "Bounds on the Dimension of Spaces of Multivariate Piecewise Polynomials," *Rocky Mountain Journal of Mathematics* **14**, pp. 251–264.

[Si78] Sibson, R. "Locally Equiangular Triangulations," *The Computer Journal* **21**, pp. 243–245.

[Z70] Zenisek, A. "Interpolation Polynomials on the Triangle," *Numer. Math.* **15**, pp. 283–296.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2937	2. GOVT ACCESSION NO. AD-A172776	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ITERATIVE METHOD FOR COMPUTING BIVARIATE C^1 PIECEWISE CUBIC POLYNOMIAL INTERPOLANTS		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Thomas A. Grandine		8. CONTRACT OR GRANT NUMBER(s) DAAG29-80-C-0041 DMS-8210950, Mod. 4
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of Wisconsin 610 Walnut Street Madison, Wisconsin 53705		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 3 - Numerical Analysis and Scientific Computing
11. CONTROLLING OFFICE NAME AND ADDRESS See Item 18 below.		12. REPORT DATE June 1986 13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

National Science Foundation
Washington, DC 20550

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

iterative method
SOR
piecewise polynomial
B-form of a polynomial

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper describes a successive overrelaxation (SOR) method for computing a class of bivariate C^1 piecewise cubic polynomial interpolants. Given a collection of points in R^2 together with a triangulation of those points, the schemes described require only the values of the function to be interpolated at the given points. The result is a C^1 interpolant which is a cubic polynomial over each of the triangles in the triangulation. Numerical results are presented for a typical scheme in this class.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

END

11-86

DTIC